

## Lista de exercícios – 8

Crie um novo projeto Eclipse para cada um dos exercícios abaixo. Sugere-se os seguintes nomes de projeto: *projExercicio1Lista8*, *projExercicio2Lista8*, *projExercicio3Lista8* e *projExercicio4Lista8*.

### Exercício 1 (PARA ENTREGAR)

- 1) Crie uma classe *Produto*.
  - a) A classe *Produto* deverá possuir os atributos privados *nome* e *valorUnitario*.
  - b) Crie os respectivos métodos getter e setter para todos os atributos da classe *Produto*. O método setter do atributo *valorUnitario* não deve permitir atribuir valor negativo.
  - c) Crie um método construtor para a classe *Produto*, que recebe como parâmetro dois valores, correspondentes ao nome e ao valor unitário de um produto. Estes valores de parâmetro devem ser usados para inicializar os atributos *nome* e *valorUnitario*.
  - d) Crie um segundo método construtor para a classe *Produto*, que não recebe nenhum parâmetro (método construtor padrão).
- 2) Crie uma classe *ItemCarrinhoCompras*. Esta classe modela um item de carrinho de compras, como um item de carrinho de compras de um site de comércio eletrônico.
  - a) A classe *ItemCarrinhoCompras* deve possuir os atributos privados *quantidade* e *produto*. O atributo *quantidade* deve ser do tipo *int*, enquanto o atributo *produto* deve ser do tipo *Produto*.
  - b) Crie os respectivos métodos getter e setter para cada um dos atributos da classe *ItemCarrinhoCompras*. O método setter do atributo *quantidade* não deve permitir atribuir valor negativo.
  - c) Na classe *ItemCarrinhoCompras*, crie o método público *obterValorTotal*. Este método não recebe parâmetros. O método deverá calcular e retornar o valor total do item de carrinho de compras (*quantidade* \* *valor unitário do produto*).
- 3) Crie uma classe *Programa*, que contenha um método *main*.
  - a) O método *main* deverá solicitar ao usuário que informe, via teclado, um nome e um valor unitário de um produto. Estas informações deverão ser armazenadas em uma instância (objeto) da classe *Produto*.
  - b) O usuário também deverá informar, via teclado, uma quantidade adquirida do produto informado anteriormente. Esta informação deverá ser armazenada em uma instância (objeto) da classe *ItemCarrinhoCompras*. Lembre também de guardar nesta instância de *ItemCarrinhoCompras* a instância de *Produto*, criada anteriormente.
  - c) O método *main* deverá invocar o método *obterValorTotal* do objeto *ItemCarrinhoCompras*, instanciado anteriormente, obter seu retorno e exibir na tela.

## Exercício 2 (PARA ENTREGAR)

1) Crie uma classe *Ponto*.

- a) A classe *Ponto* deve possuir os atributos privados *x* e *y*. Estes atributos correspondem às coordenadas de um ponto em um plano cartesiano.
- b) Crie os respectivos métodos públicos *getter* e *setter* para cada um dos atributos da classe *Ponto*. Os métodos *setter* não devem permitir atribuir valores negativos.
- c) Crie um método construtor para a classe *Ponto*, que recebe como parâmetro dois valores correspondentes à *x* e *y*. Este método construtor deverá inicializar os atributos *x* e *y*.
- d) Crie um segundo método construtor para a classe *Ponto*, que não recebe nenhum parâmetro (método construtor padrão).

2) Crie uma classe *Reta*.

- a) A classe *Reta* deve possuir dois atributos privados: *ponto1* e *ponto2*. Estes atributos correspondem aos pontos de uma reta e devem ser do tipo *Ponto*.
- b) Crie os respectivos métodos públicos *getter* e *setter* para cada um dos atributos da classe *Reta*.
- c) Crie um método construtor para a classe *Reta*, que não recebe nenhum parâmetro. Este método deverá criar dois objetos *Ponto* (usando o método construtor padrão) e atribuí-los aos atributos *ponto1* e *ponto2*.  

```
this.ponto1 = new Ponto();  
this.ponto2 = new Ponto();
```
- d) Crie um segundo método construtor para a classe *Reta*, que recebe como parâmetro dois valores correspondentes aos pontos da reta, inicializando os atributos *ponto1* e *ponto2*.
- e) Na classe *Reta*, crie o método público *getDistancia()*. Este método não recebe parâmetros. O método deverá calcular a distância entre os dois pontos da reta, ou seja, deverá calcular a largura da reta. Você poderá utilizar a fórmula da distância euclidiana, abaixo:

$$\sqrt{(x \text{ do ponto } 2 - x \text{ do ponto } 1)^2 + (y \text{ do ponto } 2 - y \text{ do ponto } 1)^2}$$

Para implementar a fórmula acima, você poderá precisar usar métodos estáticos da classe *Math*, demonstrados abaixo:

```
double a = Math.pow(3, 2);    // equivale a 32, com resultado 9  
double b = Math.sqrt(25);    // equivale a  $\sqrt{25}$ , com resultado 5
```

O método deverá retornar o resultado do cálculo da largura da reta.

3) Crie uma classe *Programa*, que contenha o método *main*.

- Este método deverá solicitar ao usuário que informe, via teclado, 4 valores: os valores de x e y de um primeiro ponto e os valores de x e y de um segundo ponto.
- Instancie dois objetos *Ponto*, sendo que cada objeto deverá receber os respectivos valores de x e y.
- Instancie um objeto *Reta*, utilizando o método construtor que recebe como parâmetro os dois pontos criados anteriormente.
- Faça uma chamada ao método *getDistancia()* do objeto *Reta* instanciado anteriormente, obtenha a resposta deste método e escreva o resultado na tela, como a “A distância entre os dois pontos é de X”, trocando o “X” pelo valor de resposta de *getDistancia()*.

### **Exercício 3 (PARA ENTREGAR)**

Este exercício é similar ao exercício 1 da lista 6, possuindo vários pontos em comum. A diferença é que o presente exercício considera os conceitos de relações entre classes (aula 13) e melhora a coesão da implementação, pois distribui melhor as responsabilidades das classes envolvidas.

- Crie uma classe *Atleta* com os atributos privados *idade* e *peso*.
- Crie os respectivos métodos públicos *getter* e *setter* para cada um dos atributos da classe *Atleta*. Os métodos *setter* não devem permitir a atribuição de valores negativos.
- Crie uma classe *AtletaUtil*, sem atributos.
- A classe *AtletaUtil* deverá conter um método público e estático chamado *obterCategoria*. Este método recebe 1 parâmetro: uma instância de *Atleta*. O método deverá retornar a categoria do atleta, de acordo com as definições da tabela abaixo.

Idade (em anos)	Peso (em quilos)	Categoria
até 12 (inclusive)		Infantil
13 (inclusive) a 16 (inclusive)	até 40 (inclusive)	Juvenil leve
	acima de 40	Juvenil pesado
17 (inclusive) a 24 (inclusive)	até 45 (exclusive)	Senior leve
	45 (inclusive) a 60 (inclusive)	Senior médio
	acima de 60	Senior pesado
acima de 24		Veterano

- e) Crie uma classe *Programa*, que contenha um método *main*.
- f) O método *main* deverá solicitar ao usuário que informe, via teclado, uma idade e um peso de um atleta. Estas informações deverão ser armazenadas em uma instância (objeto) da classe *Atleta*.
- g) O método *main* deverá invocar o método *obterCategoria* da classe *AtletaUtil*, obter seu retorno (a categoria do atleta) e exibir na tela.

#### **Exercício 4 (PARA ENTREGAR)**

Este exercício é similar ao exercício 2 da lista 6, possuindo vários pontos em comum. A diferença é que o presente exercício considera os conceitos de relações entre classes (aula 13) e melhora a coesão da implementação, pois distribui melhor as responsabilidades das classes envolvidas.

- a) Crie uma classe *Aluno* com os atributos privados *nomeCompleto*, *altura* e *genero*. O atributo *genero* deverá ser do tipo *int*, pois irá receber os valores 1, para masculino, ou 2, para feminino. O atributo *altura* deverá receber um valor em centímetros.
- b) Crie os métodos públicos *getter* e *setter* para cada um dos atributos da classe *Aluno*. O método *setter* do atributo *altura* não deverá permitir a atribuição de valor negativo. O método *setter* do atributo *genero* não deverá permitir a atribuição de valores diferentes de 1 e 2.
- c) Crie uma classe *AcademiaUtil*, sem atributos.
- d) Na classe *AcademiaUtil*, crie o método público e estático *obterPesoIdeal*. Este método deverá receber 1 parâmetro: uma instância de *Aluno*. O método deverá calcular e retornar o peso ideal do aluno, baseado em uma das seguintes fórmulas:  
  
$$(altura - 100) \times 0.90, \text{ se o gênero for masculino (1)}$$
$$(altura - 100) \times 0.85, \text{ se o gênero for feminino (2)}$$
- e) Crie uma classe *Programa*, que contenha um método *main*.
- f) O método *main* da classe *Programa* deverá solicitar ao usuário que informe, via teclado, um nome completo, uma altura e um gênero de um aluno de academia. Não esqueça de avisar o usuário que para o gênero, ele deverá digitar 1 para masculino ou 2 para feminino. Informe também que a altura deverá ser informada em centímetros. As informações do usuário deverão ser armazenadas em uma instância (objeto) da classe *Aluno*.
- g) O método *main* deverá invocar o método *obterPesoIdeal* da classe *AcademiaUtil*, obter o valor de retorno (peso ideal) e exibir na tela uma mensagem informando o peso ideal do aluno.